UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/522,790	01/31/2005	Stephane Arcaro	2937-127	1745

6449 7590 07/09/2009
ROTHWELL, FIGG, ERNST & MANBECK, P.C.
1425 K STREET, N.W.
SUITE 800
WASHINGTON, DC 20005

EXAMINER	
TECKLU, ISAAC TUKU	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
07/09/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTO-PAT-Email@rfem.com

Office Action Summary	**Application No.** 10/522,790	**Applicant(s)** ARCARO ET AL.	
	Examiner ISAAC T. TECKLU	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 April 2009*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *11-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *11-31* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. New claims 29-31 have been added.

2. Claims 11-31 have been examined.

Response to Arguments

3. Applicant's arguments filed 04/23/2009 have been fully considered but they are not persuasive.

Argument:

"Brassard merely discloses a single code generator able to target different languages one at a time and not several code generators, as claim 11 requires." (Remark, pages 9-10)

Response:

Examiner would like to direct the Applicant to Fig. 3 of prior art Brassard to indicate that Brassard, contrary to the above argument, discloses several code generators to translate the part of the description that the generators provide. Fig. 3 vividly illustrates several code generators being used to translate the part of the description that the generators provide. For instance, Fig. 3, 50 is used by the developer to generate the model declaration 51. The model declaration can either be language-independent or adapted for use with a specific programming language such as C++, JAVA.TM. Further, Fig. 3 illustrates the several code generators such as Model Declaration Engine 46, Generation Instruction Engine 47 and Recursion binder 48. Thus, it is respectfully submitted that the above argument is not persuasive and hence the rejection has been maintained as set forth in the Office Action.

Argument:

"The portion of Brassard cited by the Examiner fails to disclose modifiable distribution rules and Applicants can find no disclosure of modifiable distribution rules in Brassard." (Remark, page 10)

Response:

Examiner would like to indicate that Fig. 8 clearly shows amendment of the source code done by modifying the set of generation instructions. For the conveniences of the Applicant, portion of the prior art of record which describes the above concept is provided below.

> "FIG. 8 illustrates a method according to another preferred embodiment of the present invention in which source code generated by a preferred method is modified and re-generated without excessive manipulation of the code by the user. The code is retrieved 180 either in a text editor or a user-interface. The user evaluates which modifications are needed 181 in the source code. Pointers to the set of generation instructions are obtained 182. If necessary, pointers to the model declaration and the generation parameters are also obtained. A change is then made 183 to at least one of the set of generation instructions, the model declaration and the generation parameters (filter variables, context variables and number of blocks of code of each kind that were used). This change can be made automatically or manually by the user. The change made could be to change the order or re-sequence the sets of generation instructions. Finally, when appropriate changes have been made, the computer source code can be regenerated 184." (col.21:45-60, emphasis added)

> "The developer uses the code generator user interface 49 for the remaining tasks. Once he invokes it, he loads the file containing the model declarations and he loads the file containing the sets of generation instructions 56 if they were already done. In the case of the preferred embodiment, the developer creates the required number of sets of generation instructions 56 to suit his own needs (step 77). The sets of generation instructions can be created, modified, customized or ordered to prepare the set of suitable sets of generation instructions for the generation." (col.12:40-55, emphasis added)

Thus, it is respectfully submitted that the above argument is not persuasive and hence the rejection has been maintained as set forth in the Office Action.

Argument:

"Applicants can find no disclosure in Brassard's Specification relating to services that cannot be defined by the language, as claimed in claims 25 and 25."

Response:

Examiner would like to indicate that Brassard clearly discloses a software description language or software with a software description language in which the other class cannot have access to any one of technical or function service except through … first class in the following portion of the prior art.

Brassard teaches enabling to graphically or textually build a computer application model (see at least col.2:20-30 "… develop Universal Modeling Language…", col.3:15-35 "… interact textually or graphically with visual modeling tool or integrated development tool…", e.g. FIG. 1, Visual modeling tool or integrated development environment 30 and 32 and related text) and to provide a description of the model in a software description language organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application (col.3:15-35 "… interact textually or graphically with visual modeling tool or integrated development tool…", col.5:25-40 "… model declaration would include a list of classes… the description of the operations…" e.g. FIG. 1, Library of pre-defined templates , class 32, FIG. 12a-b EJB classes and related text), in which: the said services cannot be defined by said language (see at least e.g. FIG. 3, 50, 55; FIG. 11 – core services for a banking application - and related text), and the other classes cannot have access to any one of these technical or functional services except through said first classes (see at least e.g. FIG. 3, 49, 52, FIG. 12a-b EJB classes and related text). Thus, it is respectfully submitted that the above argument is not persuasive and hence the rejection has been maintained as set forth in the Office Action.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 11-31 are rejected under 35 U.S.C 102(e) as being anticipated by Brassard et al. US 6,742,175 B1 (hereinafter "Brassard").

As per Claim 11, Brassard teaches a software, encoded on a computer readable medium, for generation of a computer code of at least one part of a computer application (see at least e.g. FIG. 3, 60 and related text), in which the software generates the said computer code from a description of said at least one part of the computer application by distributing said description between several code generators according to modifiable distribution rules (see at least e.g. FIG. 3, Model declaration 51, General instruction 56, Generator option for a specific language associated with components development 57, Model Declaration Engine 45, Generation Instruction Engine 47, Recursion binder 48, FIG. 8, Change at least one of Set of Generation Instructions, Generation Parameters, Model Declaration 183 and see at least e.g. FIG. 12a-b and related text and col.3:25-35 "… sub-classing of the existing components and extending its functionality …", col.21:45-60, col.12:40-55 and col.33:55-67), each code generator translating the part of said description that it is provided with, in order to provide at least one part of the said computer code in a respective language (see at least e.g. FIG. 3, Generated source code for specific language associated with components development 60).

As per Claim 12, Brassard teaches splitting up said description in object classes (see at least e.g. FIG. 3, 49 and FIG. 5b, 86); the software distributing said object classes between the code generators according to said distribution rules (see at least col.12:57-67 "… file containing the model declarations and file containing the set of generation instruction ... generation instructions can be created..."), each code generator translating the object classes that it is provided with, into said corresponding part of the said computer code (see at least e.g. FIG. 3, Generated source code for specific language associated with components development 60).

Claim 13, Brassard teaches splitting up said description in dependencies between said object classes, and in the case of a dependency between two object classes each translated by a different code generator (see at least e.g. FIG. 12a-b and related text), the software makes said dependency be handled by two adapters that each translate it into a computer code for interfacing (see at least e.g. FIG. 3, Generation Instruction Engine 47, Recursion binder 48)the computer codes produced by said code generators for said two object classes (see at least col.33:60-67 "… for each class contained … classes that are dependent on …").

As per Claim 14, Brassard teaches which each of the two adapters produce said respective interfacing computer code for a respective object class among said two object classes (see at least e.g. FIG. 12a-b and related text and col.3:25-35 "… sub-classing of the existing components and extending its functionality …" and col.33:55-67).

As per Claim 15, Brassard teaches each of the two adapters inserts the respective interfacing computer code into the computer code produced by one of said code generators for said object class for which the adapter has produced said interfacing computer code (see at least e.g. FIG. 13e and related text).

As per Claim 16, Brassard teaches two adapters having to handle the dependency are chosen by the software following assignment rules associating (see at least e.g. FIG. 3, 55 and related text), for the orientation of the dependency of said two object classes, an adapter corresponding to each of the code generators translating said two object classes, the said assignment rules being modifiable (see at least e.g. FIG. 12a-b and related text and col.33:55-67).

As per claim 17, this is similar limitation substantially paralleling the limitation in claim 16, thus, this limitation have been addressed as set forth above.

As per claim 18, this is similar limitation substantially paralleling the limitation in claim 16, thus, this limitation have been addressed as set forth above.

As per claim 19, Brassard discloses generating said computer code from said description made in a language organized in object classes (see at least e.g. FIG. 5d, step 112-114 and related text), in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application (see at least e.g. FIG. 3, 60 and related text), said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes (see at least e.g. FIG. 5d, steps 118-122 and related text).

As per claim 20, Brassard discloses the software according to the claim 19, distributing said description between the code generators according to distribution rules associating at least some of said first classes or of said other classes of said language with at least one of said code generators (see at least e.g. FIG. 3, Model declaration 51, General instruction 56, Generator option for a specific language associated with components development 57, Model Declaration Engine 45, Generation Instruction Engine 47, Recursion binder 48).

As per claim 21, Brassard discloses the software according to claim 20, splitting up said description in object classes (see at least e.g. FIG. 3, 49 and FIG. 5b, 86), the software distributing said object classes between the code generators according to said distribution rules (see at least col.12:57-67 "… file containing the model declarations and file containing the set of generation instruction ... generation instructions can be created..."), each code generator translating the objects classes that it is provided with, into said corresponding part of said computer code and wherein the software splits up said description into first classes or into other classes of said language (see at least e.g. FIG. 3, Generated source code for specific language associated with components development 60).

As per claim 22, Brassard discloses the software according to the claim 17, generating said computer code from said description made in a language organized in object classes (see at least e.g.

FIG. 5d, step 112-114 and related text), in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language (see at least e.g. FIG. 3, 60 and related text), and the other classes of said language cannot have access to any one of said services except through said first classes and wherein the software splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language (see at least e.g. FIG. 5d, steps 118-122 and related text).

As per claim 23, Brassard discloses the software according to the claim 17, generating said computer code from said description made in a language organized in object classes (see at least e.g. FIG. 5d, step 112-114 and related text), in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes (see at least e.g. FIG. 3, 60 and related text), wherein the software distributes said description between the code generators according to distribution rules associating at least some of said first classes or of said other classes of said language with at least one of said code generators, and wherein the software splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language (see at least e.g. FIG. 5d, steps 118-122 and related text).

As per claim 24, Brassard discloses the software according to the claim 17, generating said computer code from said description made in a language organized in object classes (see at least e.g. FIG. 5d, step 112-114 and related text), in which said language enables to define first classes giving access to technical or functional services to be provided by a hardware and software computer platform receiving the computer application, said services being not definable by said language, and the other classes of said language cannot have access to any one of said services except through said first classes, wherein the software splits up said description into first classes or into other classes of said language (see at least e.g. FIG. 3, 60 and related text), wherein the software further splits up said description in dependencies between said object classes from dependencies between said first classes or other classes of said language, and wherein the software distributes said description between the code generators according to distribution rules associating at least some of said first classes or of said other classes of said language with at least one of said code generators (see at least e.g. FIG. 5d, steps 118-122 and related text).

As per claim 25, Brassard discloses a software description language,_encoded on a computer readable medium, organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which: the said services cannot be defined by said language (see at least e.g. FIG. 5d, step 112-114 and related text), and the other classes cannot have access to any one of these technical or functional services except through said first classes (see at least e.g. FIG. 5d, steps 118-122 and related text).

As per claim 26, Brassard discloses the software description language according to claim 25 of the type of an object-oriented language for computer application modeling (see at least e.g. FIG. 4, Modeling Declaration in Modeling Tool 65 and related text).

As per claim 27, Brassard discloses a software, , encoded on a computer readable medium, (see at least col.2:1-15 "… computer readable medium…"), enabling to graphically or textually build a computer application model (see at least col.2:20-30 "… develop Universal Modeling Language…", col.3:15-35 "… interact textually or graphically with visual modeling tool or integrated development tool…", e.g. FIG. 1, Visual modeling tool or integrated development environment 30 and 32 and related text) and to provide a description of the model in a software description language organized in classes enabling to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application (col.3:15-35 "… interact textually or graphically with visual modeling tool or integrated development tool…", col.5:25-40 "… model declaration would include a list of classes… the description of the operations…" e.g. FIG. 1, Library of pre-defined templates , class 32, FIG. 12a-b EJB classes and related text), in which:

the said services cannot be defined by said language (see at least e.g. FIG. 3, 50, 55; FIG. 11 – core services for a banking application - and related text), and

the other classes cannot have access to any one of these technical or functional services except through said first classes (see at least e.g. FIG. 3, 49, 52, FIG. 12a-b EJB classes and related text).

As per claim 28, Brassard discloses the software according to claim 27, enabling to graphically or textually build a model of computer application human-machine interface (see at least e.g. FIG. 3, 50, 55 and related text).

Per claim 29-31 (New), these are software, encoded on either a random access memory or hard disk claims substantially paralleling the limitations in software, encoded on computer readable medium method claims 1, 25, and 27. Brassard further discloses the use of such media (see, col.8:5-15 computer readable medium) in implementing the prescribed steps, and all other limitations have been addressed as set forth above.

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192